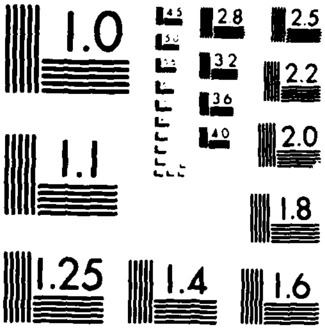
AD-A151 732 A DECOMPOSITION PROCEDURE FOR CONVEX QUADRATIC PROGRAMS 1/1
(U) GEORGIA INST OF TECH ATLANTA PRODUCTION AND
DISTRIBUTION RESEARCH CENTER C M SHETTY ET AL. JAN 85
UNCLASSIFIED PDRC-85-01 N00014-83-K-0147 F/G 12/1 NL

END

FILMED

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963 A

Georgia Institute of Technology

A DECOMPOSITION PROCEDURE FOR
CONVEX QUADRATIC PROGRAMS

by

C. M. Shetty
Mohammed Ben Daya

PDRC-85-01

PRODUCTION and
DISTRIBUTION RESEARCH
CENTER

SCHOOL OF INDUSTRIAL
AND SYSTEMS
ENGINEERING
GEORGIA INSTITUTE OF TECHNOLOGY
A UNIT OF THE UNIVERSITY SYSTEM
OF GEORGIA
ATLANTA, GEORGIA 30332

PDRC Report Series 85-01
January 1985

2

A DECOMPOSITION PROCEDURE FOR CONVEX QUADRATIC PROGRAMS

by

C. M. Shetty
Mohammed Ben Daya

PDRC-85-01

School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0205

This work was supported by the Office of Naval Research under Contract No. N00014-83-K-0147. Reproduction is permitted in whole or in part for any purpose of the U. S. Government.

ABSTRACT

A DECOMPOSITON PROCEDURE FOR CONVEX QUADRATIC PROGRAMS

C. M. Shetty
Mohammed Ben Daya
School of Industrial and Systems Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332-0205

This paper deals with the solution of convex quadratic programs by iteratively solving a master problem and a subproblem as proposed previously by Sacher. The approach has the advantage that the subproblems are linear programs so that existing schemes for solving large problems can be taken advantage of. This paper gives a closed form solution to the master problem so that the procedure is well suited for solving large quadratic programs and can take advantage of the constraint structure.

1. INTRODUCTION

Consider the following quadratic program

$$\begin{array}{lll} \text{QP:} & \text{minimize} & \frac{1}{2} x^t Bx + c^t x \\ & \text{subject to} & Ax > b \\ & & x > 0 \end{array}$$

where $x \varepsilon R^n$, $A \varepsilon R^{m \times n}$ and B is positive semi-definite.

A decomposition algorithm using Lemke's method was proposed by Sacher [1] to solve Problem QP. The method solves the original problem by iteratively solving a master program and a subprogram.

In this paper, we solve the master program using the gradient projection method of Rosen [2] in Section 3 and by the Reduced Gradient Algorithm in Section 5. The methods take advantage of the special structure of the master program and reduce the algorithms to simple arithmetic calculations.

2. The Sacher Algorithm

Sacher's procedure uses an inner linearization approach followed by restriction and is a nonlinear version of the Dantzig-Wolfe decomposition principle [3]. His algorithm is essentially of von Hohenbalken's simplicial decomposition method [4] [5] when specialized to a quadratic objective function.

Algorithm:

Step 0:

Let P and Q be matrices whose columns are affinely independent extreme points and directions respectively of the set $\{x: Ax > b, x > 0\}$. Here P must have at least one column, but Q may be vacuous.

Step 1:

(Master program). Solve the following problem by Lemke's method.

$$\text{MP:} \quad \text{minimize} \quad \frac{1}{2}\binom{u}{v}^t \bar{B}\binom{u}{v} + \bar{c}\binom{u}{v}$$

$$\text{subject to} \quad e^t u = 1; \quad u > 0,\ v > 0$$

where $\bar{B} = (P,Q)^t B(P,Q)$ and $\bar{c} = \binom{P^t c}{Q^t c}$. Denote the solution by $[\bar{u},\bar{v}]$. If the solution is unbounded, STOP. The original problem has an unbounded objective function.

Step 2:

Set $\bar{x} = P\bar{u} + Q\bar{v}$.

Step 3:

(Subprogram). Solve the following program.

$$\text{SP:} \quad \text{minimize} \quad h^t x$$

$$\text{subject to} \quad Ax > b,\ x > 0$$

where $h = BP\bar{u} + BQ\bar{v} + c$. If the solution is bounded denote it by $\hat{x}$. Otherwise, let $\hat{x}$ be a (normalized) direction of recession for which $h^t\hat{x} < 0$.

Step 4:

Stop if $\hat{x}$ is an extreme point and $h^t\bar{x} = h^t\hat{x}$,the solution to QP is $\bar{x}$; otherwise go to Step 5.

Step 5:

(Updating). If $u_i = 0$ ($v_i = 0$, respectively), delete column i of P (Q, respectively). If x is an extreme point, replace P by $(P,\hat{x})$. Otherwise replace Q by $(Q,\hat{x})$. Go to Step 1.

3. Solving the Master Program Using the Gradient Projection Method

We will solve the master program MP using the Gradient Projection Method (GPM) of Rosen [2]. It may be recalled that at a feasible point, the GPM method projects the gradient of the objective function on the null space of the gradients of the binding constraints. It will find an improving feasible direction, if one exists, and a one dimensional optimization problem is solved to find the optimum distance to move in the chosen direction. Because of the special structure of the master problem, closed form solutions to the above can be obtained, and the resultant algorithm is given below. Section 4 shows that the steps specified are precisely those obtained by the GPM method.

Algorithm:

Step 0:

Choose a point (u_1,v_1) with $e^t u_1 = 1$, $u_1 > 0$ and $v_1 > 0$.

Let $I_p = \{i: \ u_i > 0\}$; $|I_p| = n_p$

$I_d = \{i: \upsilon_i > 0\}$; $|I_d| = n_d$; $n = n_p + n_d$

$K_p = \{i: u_i = 0\}$; $\bar{K}_p = I_p \backslash K_p$ $|K_p| = p$

$K_d = \{i: v_i = 0\}$; $K_d = I_d \backslash K_d$; $|K_d| = d$

Let $k = 1$ and go to step 1.

Step 1:

Compute $\nabla f(u_k,v_k)$

where $$f(u_k v_k) = \frac{1}{2} \binom{u_k}{v_k}^t \bar{B} \binom{u_k}{v_k} + \bar{c} \binom{u_k}{v_k}$$

Let $\nabla f(u_k,v_k) = (\nabla_k^1, \nabla_k^2, \ldots, \nabla_k^n)$

Step 2:

Computer d_k as follows:

$$d_k^j = \begin{cases} 0 & \text{if } j \in K_p \cup K_d \\ \alpha_k - \nabla_k^j & \text{if } j \in \bar{K}_p \\ \nabla_k^j & \text{if } j \in \bar{K}_d \end{cases}, \quad j = 1,2,\ldots,n \tag{1}$$

where $\alpha_k = \sum_{j \in \bar{K}_p} \left(\frac{1}{n_p - p}\right) \nabla_k^j$

If $d_k \neq 0$ go to step 3.

If $d_k = 0$, compute w as follows:

$$w_0 = - \sum_{j \in \bar{K}_p} \left(\frac{1}{n_p - p}\right) \nabla_k^j \tag{2}$$

$$w_i = \begin{cases} \nabla_k^i - \sum_{j \in \bar{K}_p} \left(\frac{1}{n_p - p}\right) \nabla_k^j & \text{if } i \varepsilon K_p \\ \\ \nabla_k^i & \text{if } i \varepsilon K_d \end{cases} \tag{3}$$

If $w_i > 0$, the current solution is optimal to MP. Otherwise, let $w_j = \min_{i \varepsilon K_p \cup K_d} \{w_i\}$. Update K_p and K_d by deleting the index j, and repeat

Step 3:

Let $\lambda_k = \min \{\lambda^*, \lambda_{max}\}$

where

$$\lambda^* = - \frac{2\binom{u_k}{v_k}^t \bar{B} \; d_k + \bar{c} \; d_k}{2 d_k^t \; \bar{B} \; d_k} \tag{4}$$

and λ_{max} the maximum value of λ such that $(u_k, v_k) + \lambda d_k$ is nonnegative, clearly $\lambda_{max} = \infty$ if $d_k \geq 0$ and $\lambda_{max} = \min\{\lambda^u, \lambda^v\}$ if $d_k \ngeq 0$ where

$$\lambda^u = \min\{-u_k^j/d_k^j : \quad j \in \bar{K}_p,\ d_k^j < 0\}$$

$$\lambda^v = \min\{-v_k^j/d_k^j : \quad j \in \bar{K}_d,\ d_k^i < 0\}$$

If $\lambda_k = \infty$, the original problem is unbounded. Otherwise, let

$$\binom{u_{k+1}}{v_{k+1}} = \binom{u_k}{v_k} + \lambda_k d_k$$

Update K_p, K_d

Replace k by k+1 and go to Step 2.

It may be noted that at step 0 of iteration 2 onwards of the main algorithm, one could use the optimal of the previous iteration as the starting point. Also, in solving the master problem, at each iteration only K_p and K_d need be updated at Step 2. At Step 3, in computing λ^* it may be convenient to update $\bar{B} = (P,Q)^t B(P,Q)$ as follows.

Consider the case where the subprogram yields an extreme point $\bar{p}$. We must introduce a new row and column to $\bar{B}$. Denoting the column of P(Q) by $P_i(Q_i)$, it is easily verified that the new matrix is

$$\left[\begin{array}{c:c} \bar{B} & \alpha \\ & \\ \hdashline \beta & \theta \end{array}\right]$$

where

$$\alpha_i = \begin{cases} P_i^t B \bar{p} & \text{if } i \varepsilon I_p \\ Q_i^t B \bar{p} & \text{if } i \varepsilon I_d \end{cases}$$

$$\beta_i = \begin{cases} \bar{p}^t B P_i & \text{if } i \varepsilon I_p \\ \bar{p}^t B Q_i & \text{if } i \varepsilon I_d \end{cases}$$

$$\theta = \bar{p}^t B \bar{p}$$

In the following section, we give a justification to the algorithm based on the gradient projection method of Rosen and using the structure of the master program. For more details about the gradient projection method see [6].

4. Justification of the procedure

Let M be the matrix of the coefficients of the binding constraints. M has the following form:

$$\left[\begin{array}{c|c} e^t & 0 \\ \hline D & 0 \\ \hline 0 & C \end{array}\right]$$

where e^t is a $1 \times n_p$ vector of ones, D is a $p \times n_p$ matrix and $p = |K_p|$. Each row in D has zeros everywhere except a -1 in the jth column where $j \epsilon K_p$. C is a $d \times n_p$ matrix where $d = |K_d|$. Each row in C has zeros everywhere except a -1 in the jth column where $j \epsilon K_d$.

It is easy to show that $(M^t M)^{-1}$ is as follows:

$$\left[\begin{array}{c|c|c} \frac{1}{n_p - p} & \left(\frac{1}{n_p - p}\right) e^t & 0 \\ \hline \frac{1}{n_p - p}\, e & I_{n_p} + \frac{1}{n_p - p}\, E & 0 \\ \hline 0 & 0 & I_{n_d} \end{array}\right]$$

where E is an $n_p \times n_p$ matrix with ones everywhere, and I_i is an $i \times i$ identity matrix.

Then it can easily be verified that $d_k = -P\nabla f(u_k, v_k) = -\,[I - M^t(MM^t)^{-1}M]\ \nabla f(u_k, v_k)$ is given by (1), and $w = (MM^t)^{-1} M \nabla f(u_k, v_k)$ is as given by (2) and (3). Let $\psi(\lambda) = f\left(\begin{pmatrix} u_k \\ v_k \end{pmatrix} + \lambda d_k\right)$.

Solving $\frac{d\psi(\lambda)}{d\lambda} = 0$ for λ gives λ^* of Eq. (4).

5. A Reduced Gradient Algorithm

We now present a reduced gradient algorithm [6] to solve the master problem MP.

Algorithm

Step 0:

Choose a point $y_1 = \begin{pmatrix} u_1 \\ v_1 \end{pmatrix}$ satisfying

$$e^t u_1 = 1$$

$$y_1 > 0$$

Let $k = 1$ and go to Step 1.

Step 1:

Let $d_k^t = (d_B^t, d_N^t)$ where

Let i_k be the index of a positive component of $\binom{u_k}{v_k}$ (which will be the current basic variable). Let $\bar{B}_{i_k}$ be the i_kth row of $\bar{B}$ and $r^t = \bar{B}y + \bar{c} - \bar{B}_{i_k} e^t$.

The components of d_N are given by

$$d_j = \begin{cases} -r_j & \text{if } j \neq i_k \text{ and } r_j < 0 \\ -y_j r_j & \text{if } j \neq i_k \text{ and } r_j > 0 \end{cases}$$

$$d_B = d_{i_k} = -e^t d_N = -\sum_{j \neq i_k} d_j$$

If $d_k = 0$, stop; y_k is a Karush-Kuhn-Tucker point; Otherwise go to Step 2.

Step 2:

Compute

$$\lambda^* = -\frac{2y_k \bar{B} d_k + \bar{c}^t d_k}{2d_k^t \bar{B} d_k}$$

$$\text{Let } \lambda_k = \min\{\lambda^*, \lambda_{\max}\}$$

where:

$$\lambda_{\max} = \begin{cases} \min\limits_j \{\frac{-y_{jk}}{d_{jk}} : d_{jk} < 0\} & \text{if } d_k \ngtr 0 \\ \infty & \text{if } d_k > 0 \end{cases}$$

and y_{ik}, d_{jk} are the jth components of y_k and d_k respectively. If $\lambda_k = \infty$, the original problem is unbounded. Otherwise let $y_{k+1} = y_k + \lambda_k d_k$, and let k = k + 1, and return to step 1.

The reader may note that the procedure for updating $\bar{B}$ at each main iteration of the algorithm discussed in Section 3 holds in this case also.

Summary

We have presented above two algorithms to solve the master problem, Although no proof is available for the convergence of the Gradient Projection Method, it is generally recognized that the method converges. For the Reduced Gradient Algorithm, the reader may refer to [6] for a convergence proof. Finite convergence of the main iteration of Sacher's algorithm is obvious. At present computational testing is on hand to compare the effectiveness of the proposed procedures. Other schemes for special cases of the problem are also being developed.

References

1. R. S. Sacher, 1980. "A Decomposition Algorithm for Quadratic Programming," Mathematical Programming, 18, pp. 16-30.

2. J. B. Rosen, 1960. "The Gradient Projection Method for Nonlinear Programming, Part I, Linear Constraints," SIAM J. Applied Mathematics, 8, pp. 181-217.

3. G. B. Dantzig and P. Wolfe, 1960. "The Decomposition Algorithm for Linear Programming," Operations Research, 8, pp. 101-111.

4. B. von Hohenbalken, 1975. "A Finite Algorithm to Maximize Certain Pseudoconcave Functions Polytopes," Mathematical Programming, 9, pp. 189-206.

5. B. von Hohenbalken, 1977. "Simplicial Decomposition in Nonlinear Programming Algoriths," Mathematical Programming, 13, pp. 49-68.

6. M. S. Bazaraa and C. M. Shetty, 1979. Nonlinear Programming: Theory and Algorithms, John Wiley & Sons.